Real Time over ATM and IP

Z.L. Budrikis,
ATRI/CRC-BTN
Curtin University of Technology, Perth Western Australia
Presented at SPARTAN Symposium, University of Kansas, May 19-20, 1998

Outline

- Introduction
  - Requirements of real time
  - The Principle of delay constancy
- Real Time over ATM
  - CBR/DBR, AAL1
  - VBR.rt/SBR.1
- Real Time over IP
  - Packetisation delay
  - Routing delays
- Time Aware Routing

Introduction

- Requirements on real time transfer:
  - Required to provide timely transfer of information signal
    - with acceptable incidence of error and loss
  - Required to provide adequate transfer of application clock
    - presentation at receiver must be able to synchronise with application at source
- Principle of Delay Constancy:
  - Delay between instant at which real time signal is generated at source and the instant at which the signal is presented at receiver, is a constant over the duration of the session
  - The constant delay must not be larger than the bound on delay specified for the application

Introduction

- Requirements for real time signal transfer reduce to:
  - Guarantee of bandwidth for duration of session
  - Guarantee of delivery within specified time
  - Timely transfer of synchronising protocol information
- Requirements for real time are readily met if
  - Application generates information bits at constant rate AND
  - The constant rate stream is transferred in STM
- Transfer of real time signals poses problems if
  - Bit stream at source is variable rate
  - Transfer is in asynchronous mode
  - There are variable queuing, routing, *et alii*

Introduction

- Real time over ATM or IP?
  - Transfer of real time over ATM may pose problems
    - Problems are trivial if application is constant rate and transfer is on DBR capability
    - Problems range from less trivial to insurmountable if source rate is variable and capability is to be VBR-rt or SBR.1
  - Transfer of real time over IP poses bigger problems
    - More difficult to meet maximum delay specification
      - IP needs to be connection-oriented
      - Transfer of low rate applications may be inefficient
    - May require time-aware routing - However that is not

Real time over ATM

- Little or no problem if real time is CBR
  - If application has rate *R* bits per second
    - Packet delay is 47x8/*R*=376/*R* seconds
      - For 64 Kbps voice, packet delay is 5.875 milliseconds
      - Queuing delays are small by comparison
      - Propagation delay same as for STM
  - AAL1 maybe used with all AAL1 functionality
    - AAL1 includes residual time stamp (RTS):
      - allows transfer of timing information from source to receiver
  - All requirements for transfer of real time can be met

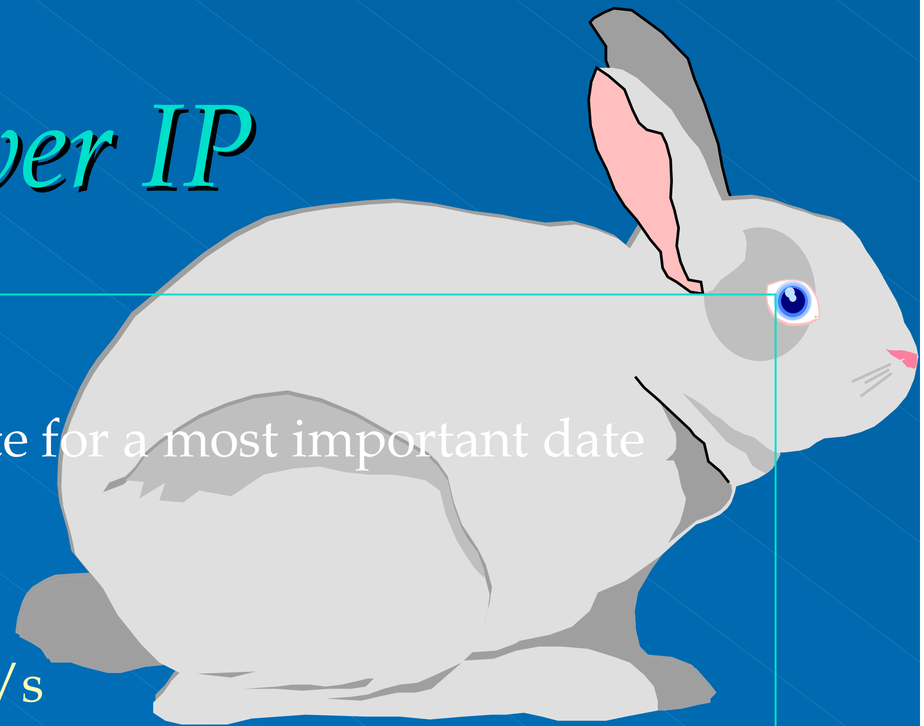
Real time over ATM

- More problem if real time signal has variable rate
  - Transfer delay may be variable and excessive
    - Else have to guarantee bandwidth equal to peak rate, making VBR more expensive than CBR
    - Packetisation delay variable
  - Principle of delay constancy requires that constant delay be equal to global maximum delay over session
  - No facility for transferring timing information from source to destination
    - AAL1 not useful for VBR - timing transfer by RTS requires constant cell rate

Real Time over IP

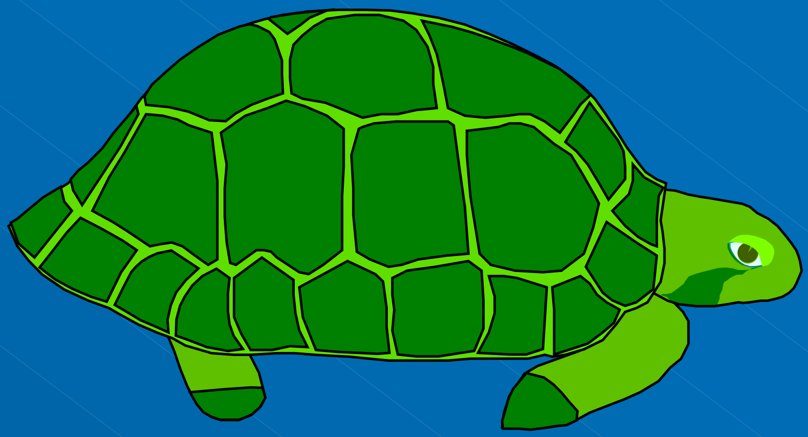
- Now, here's a challenge!
  - Real time not possible by "genuine" IP datagrams:
    - No guarantee of bandwidth
      - It works when it works, and it doesn't when it doesn't
  - Requires fixed route for session, and reserved bandwidth along route:
    - Will RSVP help? Yes, coupled to CAC, it will help
    - Effectively, will then provide connection-oriented transfer
  - Reserved bandwidth is **necessary**, but is **not sufficient**
    - Transfer must be timely, observe delay bound, and

Real Time over IP



- Delay issue
  - I’m late, I’m late for a most important date
- Packetisation delay
  - Packet pay load L bytes
  - Application rate R_1 bits/s
  - Packetisation delay 8xL/R_1 seconds
  - Depending on router operation, packetisation delay could be repeated more than once in end -to-end transfer

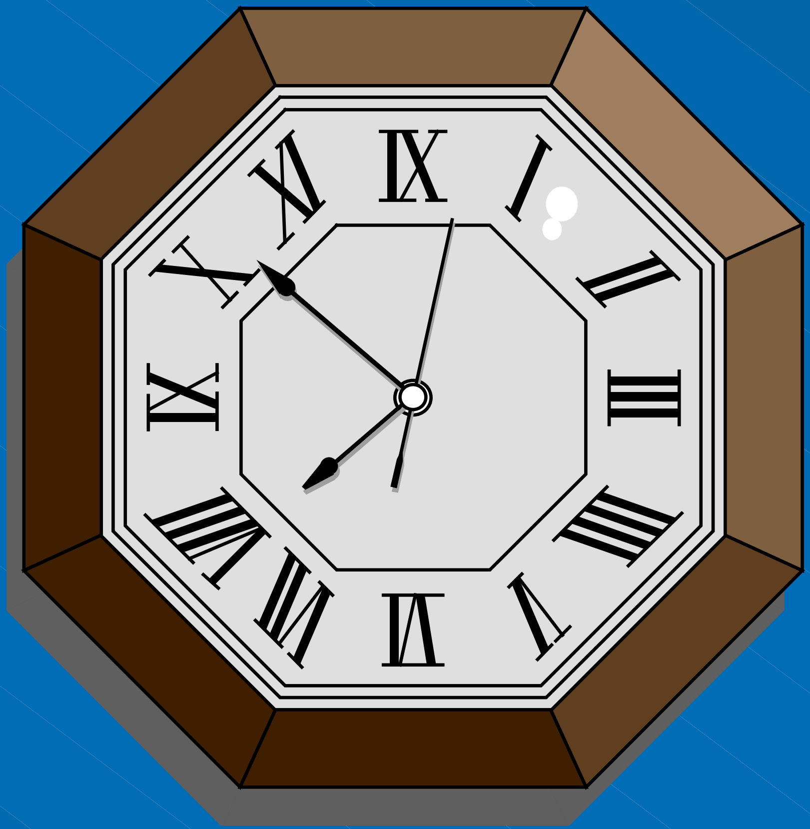
Real Time over IP



- Link delay
  - Link rate R_2 bits/s
  - Link transfer delay 8xL/R_2 seconds
  - With store and forward of packets, link transfer delay will occur as many times as the number of links on the connection
- Router delay
  - Delay through router is variable, depending on load on router

Real Time over IP

- Delay can be large and unpredictable
  - Decidedly, not what real time transfer requires
- Once presentation has commenced, any excessively delayed packet causes a discontinuity and increase in delay, or discontinuity and loss in signal



Time-Aware IP Routing *(Patent applied)*

- Assume IPv6 Internet and (as yet to be defined) header extension option, designed to support real time transfer
  - Real time signal would be sent as fragments of an IP packet or of a flow of IP packets, all sent over the same Internet connection
  - On receiving a real time packet fragment, an IP Router must forward it on time, or sooner
    - The time by which a fragment must be forwarded, is determined by the local state of the connection that carries the fragment
    - The router is time-aware with respect to all connections that are set up over it

Time-Aware IP Routing

- Time-awareness of a router is possible only if
  - the router has a clock
  - the clocks of routers and of sources/destinations are synchronous, or at least plesiochronous, with each other
  - the router is provided by sources with appropriate time information
- Assume that time information is carried in the real time header extension, and assume that this includes
  - *deltaT*, an integer that represents the time interval of the real time signal that is covered by the payload of the segment
  - *absT*, an integer that represents the time instant at the source at which the segment was completed

Time-Aware IP Routing

- Procedure at router
  - A connection "*i*", identified by source and destination IP addresses and UDP Port numbers and/or by flow label, sends fragments F_{ij} (j = 0, 1, 2, …). The total lengths of the fragments are L_{ij} bits, and the timing parameter values carried by them are $deltaT_{ij}$ and $absT_{ij}$ nanoseconds
    - 1. On receipt of F_{i0} (the first fragment), router
      - waits $WAIT_i$ nanoseconds before dispatching fragment ($WAIT_i$ is a parameter that is fixed at connection set-up);
      - notes the time t at which dispatch of F_{i0} was complete and calculates $Dl_{i1} = t + deltaT_{i0}$, the deadline completion of dispatch of the next fragment (F_{i1};

Time-Aware IP Routing

- records $T_{i\_last} = absT_{i0}$

- 2. On receipt of F_{ij} ($j > 0$), router
  - checks that $T_{i\_last} + deltaT_{ij} >= absT_{ij}$
  - if yes,
    - schedules start of dispatch $startT_i = Dl_{ij} - L_{ij}/R_i$, where R_i is the rate of link on which F_{ij} is to be sent
    - dispatches F_{ij} as soon as possible, not waiting for $startT_i$, but observes 'earliest start - next to go' rule
    - calculates $Dl_{i(j+1)} = Dl_{ij} + deltaT_{ij}$, (the next deadline)
    - records $T_{i\_last} = absT_{ij}$
    - returns to 2.
  - if no,
    - return to 1. I.e processes F_{ij} as F_{i0} (a first

Time-Aware IP Routing

- Procedure at receiver
  - Assume that fragment payloads can be independently decoded. (Do not require decoded signal from $F_{i(j-1)}$ to decode F_{ij})
    - 1. On receipt of F_{i0} (the first fragment), receiver
      - notes the time t_s, and starts presentation without delay;
      - on finish of presentation of F_{i0}, notes time t_f;
        - calculates $delta\_t = t_f - t_s$
        - if $delta\_t > deltaT_{i0}$, application clock rate is appropriately increased;
        - if $delta\_t < deltaT_{i0}$, application clock rate is appropriately decreased

Time-Aware IP Routing

- 2. On receipt of F_{ij} ($j > 0$), receiver
  - checks whether presentation buffer is empty
  - if yes,
    - returns to 1(restarts presentation following interrupt)
  - if no,
    - waits for presentation of previous fragment to complete;
    - returns to 1 (continues presentation without interrupt)

Time-Aware IP Routing

- Expected Performance
  - With proper design and CAC, interruptions in presentation at receiver can be expected to be rare:
    - It can be expected that all routers, including immediately upstream router, will complete dispatch of F_{ij} by their deadline Dl_{ij}
      - Therefore arrival of F_{ij} ($j > 0$) at a router can be expected to be ahead of Dl_{ij} by at least $WAIT_i$
      - $WAIT_i$ must be chosen so that with given CAC, it would be sufficient to ensure low probability of missing deadline
        - $WAIT_i$ must not be larger than necessary. It is a component in the end-to-end delay

Time-Aware IP Routing

- End-to-end delay:
  - Assuming no interrupts to presentation at receiver, fixed end-to-end delay is given by the sum of three deterministic components:
    - packetisation delay $deltaT_{i0}$ of the first fragment
    - sum of $WAIT_i$ over all routers, that connection "*i*" traverses
    - end-to-end propagation delay
  - For low rate applications, packetistion delay will dominate
    - Note that first fragment is critical
  - Sum of $WAIT_i$ will dominate for high rate applications
    - Can be kept low only by fast routing and modest network loading

Time-Aware IP Routing

- Ramp-up of fragment size
  - Packetisation delay given by deltaT of first fragment
    - Possible to make deltaT - and payload size - of first fragment small enough not to exceed allowed packetisation delay
    - Subsequent fragments can be progressively increased by delta-delta_T increments until steady-state fragment size - and/or deltaT - is reached
      - Provided delta-delta_T is small compared to WAIT (say, no more than 30% of WAIT), routers will, with reasonably high probability, still be able forward before deadline, and maintain presentation continuity at destination
  - Ramp-up facility important for efficient, yet

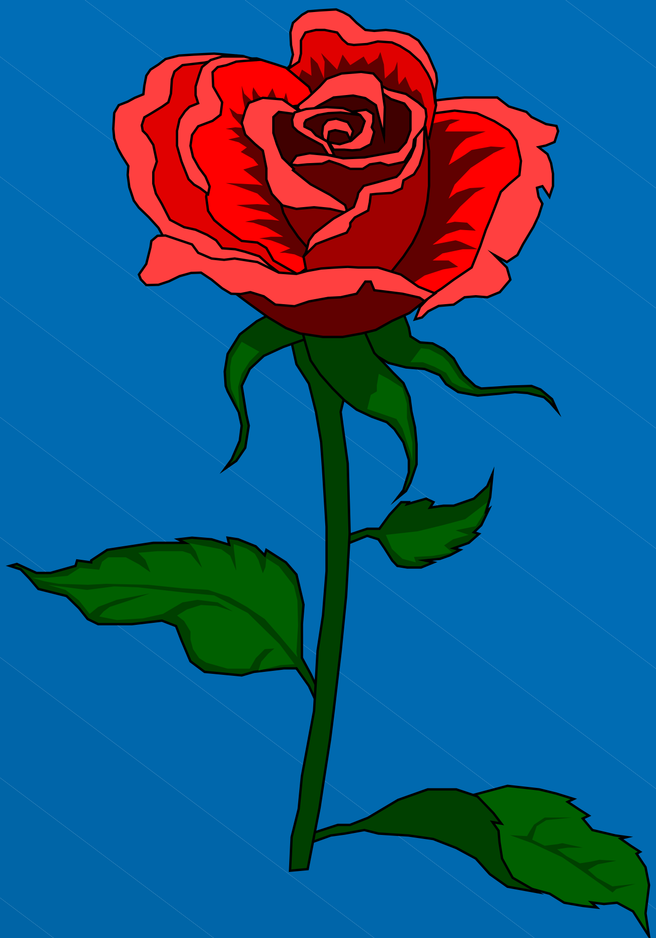
Time-Aware IP Routing

- Cross- and multi-media synchronisation
  - Two or more real-time signals from same source sent in separate streams, can be synchronised to each other for presentation at receiver:
    - $absT_{il}$ -$deltaT_{ij}$ = start of F_{ij} in stream "*i*" at source
    - $absT_{kl}$ -$deltaT_{kl}$ = start of F_{kl} in stream "*k*" at source
    - If start of F_{kl} is later than start of F_{ij} by $Later_{ki}$, then if presentation of "*i*" is started at time *t*, presentation of "k" is started at time $t + Later_{ki}$
  - Example: Compressed video is sent on one stream ("*i*"), sound on another stream ("*k*")
    - Sound and video can be mutually synchronised (lip sync can be assured) at all times, including after accidental packet loss in one or the other stream

Conclusion

- Principal consideration in real time communication:
  - (constant) end-to-end delay of signal
- Subsidiary considerations:
  - transfer of applications clock
  - synchronisation of multiple media
- Requirement is for timely transfer of signal(s) for which
  - guaranteed bandwidth is necessary
  - guarantee of bandwidth is not sufficient, unless transfer is fine-grained and regular
    - Bandwidth guarantee is sufficient for STM and CBR/DBR ATM

Conclusion



- No complete solution as yet known for transfer of variable rate real time signals over ATM
  - There is no AAL for VBR that would support synchronisation of applications' clock or of multi-media
- Complete solution for transfer of real time signals over IP is possible
  - Requires definition of real time IP header extension option